

**Gölhisar Uygulamalı Bilimler Yüksekokulu**

**NESNE TABANLI PROGRAMLAMA II DERSİ**

**PROJE KONUSU: Kanser Teşhisi ve Sınıflandırması**

**Öğrenci Ad-Soyad/Numara**

1. **Furkan Arda Karabulut - 2112903047**
2. **Bünyamin Kavraz - 2112903028**
3. **Yusuf Yeşiltepe - 2112903033**
4. **Sena Firdes Zeybek - 2212903062**
5. **Nursema Kaya - 2212903033**

MAYIS 2024 BURDUR

# GİRİŞ

Bu projede, meme kanseri teşhisi için K-En Yakın Komşu (K-NN), K-Ortalamalar (K-Means) ve Stokastik Gradyan İniş (SGDClassifier) gibi makine öğrenimi modelleri kullanılarak verilerin analiz edilmesi amaçlanmıştır. Meme kanseri teşhisi, erken evrede hayat kurtarıcı olabileceği için bu alanda etkili ve doğru modellerin geliştirilmesi büyük önem taşımaktadır. Kullanılan veri seti breast\_cancer.csv dosyasından alınmış ve çeşitli veri işleme adımları ile analiz edilmiştir.

Kullanılan veri seti, Wisconsin Breast Cancer dataset olarak bilinen ve meme kanseri teşhisinde yaygın olarak kullanılan bir veri kümesidir. Veri seti, her biri 10 farklı özellik içeren hücresel örneklerden oluşmaktadır. Bu özellikler arasında hücre boyutu, şekli ve yapısı gibi önemli faktörler bulunmaktadır.

Makine öğrenimi modelleri, sınıflandırma ve kümeleme yöntemleriyle verileri analiz ederek kanser türlerini ayırt etmede yardımcı olmuştur. K-NN modeli, benzer veri noktalarına dayalı olarak sınıflandırma yaparken, K-Means algoritması verileri kümeler halinde gruplandırmıştır. SGDClassifier ise büyük veri setleri üzerinde doğrusal sınıflandırma yaparak etkili bir performans sergilemiştir.

**Genel Bilgiler**:

***Veri Kümesi Adı***: Breast Cancer Wisconsin (Original)

***Veri Kümesinin Kaynağı***: University of Wisconsin Hospitals, Madison

***Oluşturan***: Dr. William H. Wolberg

***Veri Seti Türü***: İkili sınıflandırma

### **Veri Kümesinin Özellikleri**:

***Özellik Sayısı***: 10

***Örnek Sayısı***: 684

***Sınıf Sayısı***: 2 (Benign - iyi huylu, Malignant - kötü huylu)

***Eksik Veri***: Var (16 örnekte bir veya daha fazla özellik eksik)

### **Özellikler**:

Her örnek, bir dizi biyopsi ile elde edilen hücresel özelliklerle tanımlanmıştır. Özelliklerin tamamı tamsayı değerlerinden oluşur ve aşağıdaki gibidir:

1. ***Sample code number***: ID numarası
2. ***Clump Thickness***: Hücre kümelerinin kalınlığı
3. ***Uniformity of Cell Size***: Hücre boyutunun tekdüzeliği
4. ***Uniformity of Cell Shape***: Hücre şeklinin tekdüzeliği
5. ***Marginal Adhesion***: Marjinal yapışma
6. ***Single Epithelial Cell Size***: Tek epitel hücresi boyutu
7. ***Bare Nuclei:*** Çıplak çekirdekler
8. ***Bland Chromatin:*** Soluk kromatin
9. ***Normal Nucleoli***: Normal nükleoller
10. ***Mitoses***: Mitoz sayısı

### **Sınıflar**:

***Benign (iyi huylu)***: Sınıf 2 olarak kodlanmıştır.

***Malignant (kötü huylu)***: Sınıf 4 olarak kodlanmıştır.

### **Veri Kümesi Kullanımı**:

Bu veri kümesi, özellikle makine öğrenimi algoritmalarının eğitim ve test edilmesi için yaygın olarak kullanılmaktadır. Amaç, biyopsi sonuçlarına dayanarak bir tümörün iyi huylu mu yoksa kötü huylu mu olduğunu belirlemektir. Bu veri kümesi, çeşitli sınıflandırma algoritmalarının doğruluğunu ve etkinliğini değerlendirmek için de kullanılabilir.

### **Erişim ve İndirme:**

Veri kümesine UCI Makine Öğrenimi Deposundan erişilebilir ve indirilebilir. İşte doğrudan bağlantı: [Breast Cancer Wisconsin (Original) Data Set](https://archive.ics.uci.edu/dataset/15/breast+cancer+wisconsin+original)

# GEREÇ VE YÖNTEM

**K-Nearest Neighbors (KNN) Algoritması:**

**1. Tarihsel Geçmiş**

K-Nearest Neighbors (K-NN) algoritması, ilk kez 1951 yılında Evelyn Fix ve Joseph Hodges tarafından "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties" adlı çalışmalarında tanıtılmıştır. Bu çalışma, istatistiksel karar teorisi bağlamında sınıflandırma problemlerine odaklanmıştır. KNN, basit ve sezgisel yapısıyla popüler hale gelmiş, zamanla birçok farklı alan ve uygulamada geniş bir şekilde kullanılmıştır.

**2. KNN Algoritmasının Temelleri**

K-NN, denetimli öğrenme algoritmalarından biridir ve hem sınıflandırma hem de regresyon problemlerinde kullanılır. Temel prensibi, bir veri noktasının sınıfını veya değerini, en yakın ‘*k*’ komşusunun sınıflarına veya değerlerine göre belirlemektir**.**

İşte K-NN algoritmasının temel adımları:

**Mesafe Hesaplama:**

Test verisi ile tüm eğitim verileri arasındaki mesafe hesaplanır. En yaygın kullanılan mesafe metrikleri Euclidean, Manhattan ve Minkowski mesafeleridir.

**Komşuları Belirleme:**

En yakın 𝑘 komşusu seçilir.

**Sınıflandırma/Regresyon:**

Sınıflandırma için, en yakın komşuların sınıflarına göre çoğunluk sınıfı seçilir. Regresyon için ise, en yakın komşuların ortalaması alınır.

**3. Kullanım Alanları**

KNN algoritması, aşağıdaki alanlarda ve uygulamalarda yaygın olarak kullanılır:

1. *Metin Madenciliği***:** Dokümanların sınıflandırılması ve bilgi geri getirme sistemleri.
2. *Tıp:* Hastalık teşhisi ve tıbbi görüntü sınıflandırması.
3. *Pazarlama:* Müşteri segmentasyonu ve tavsiye sistemleri.
4. *Finans:* Kredi riski analizi ve dolandırıcılık tespiti.
5. *Görüntü İşleme:* Nesne tanıma ve yüz tanıma.

**4. Avantajlar ve Dezavantajlar**

**Avantajlar:**

1. *Basit ve Kolay Anlaşılır*: KNN, temel ve sezgisel bir algoritmadır. Karmaşık matematiksel modellemeler gerektirmez.
2. *Parametrik Olmayan*: Veri dağılımı hakkında herhangi bir varsayım yapmaz.
3. *Esneklik*: Hem sınıflandırma hem de regresyon problemlerinde kullanılabilir.

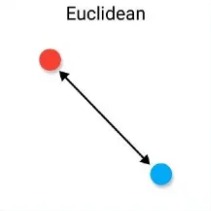
**Dezavantajlar:**

1. *Hesaplama Maliyeti*: Büyük veri setlerinde, tüm mesafeleri hesaplamak zaman alıcı olabilir.
2. *Hafıza Kullanımı*: Eğitim verilerinin tamamını hafızada tutmak gerekir.
3. *Özellik Ölçekleme*: Özelliklerin farklı ölçeklerde olması, mesafe hesaplamalarını etkileyebilir ve sonuçları bozabilir.

**5. KNN Algoritması ile İlgili Görseller ve Şekiller**

Aşağıda, KNN algoritmasını ve temel bileşenlerini açıklayan ve K-NN algoritmalarında sıklıkla kullanılan uzaklık hesaplamaları, açıklamalarıyla birlikte bazı görseller ve şekiller yer almaktadır.

### **Öklidyen Uzaklık (Euclidean Distance):**



Öklidyen uzaklık, en yaygın kullanılan uzaklık ölçüsüdür ve iki nokta arasındaki düz çizgi mesafesini hesaplar. İki nokta arasındaki Öklidyen uzaklık şu şekilde hesaplanır: 𝑑(𝑥,𝑦)=  
∑𝑖=1𝑛(𝑥𝑖−𝑦𝑖)2*d*(*x*,*y*)=∑*i*=1*n*​(*xi*​−*yi*​)2​ Burada 𝑥*x* ve 𝑦*y*, n boyutlu uzaydaki iki vektördür ve 𝑥𝑖*xi*​ ve 𝑦𝑖*yi*​, bu vektörlerin bileşenleridir​ ([IBM - United States](https://www.ibm.com/topics/knn))​​ ([FreeCodeCamp](https://www.freecodecamp.org/news/k-nearest-neighbors-algorithm-classifiers-and-model-example/" \t "_blank))​.

### **Hamming Uzaklığı (Hamming Distance):**

ekran görüntüsü, diyagram, çizgi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Hamming uzaklığı, iki aynı uzunluktaki dizi arasındaki farklı bit sayısını ölçer. Özellikle kategorik veriler ve ikili diziler için kullanılır. İki dizi arasındaki Hamming uzaklığı şu şekilde hesaplanır: 𝑑(𝑥,𝑦)=∑𝑖=1𝑛1(𝑥𝑖≠𝑦𝑖)*d*(*x*,*y*)=∑*i*=1*n*​**1**(*xi*​≠*yi*​) Burada 1**1**, şart sağlandığında 1 döner, aksi halde 0 döner​ ([IBM - United States](https://www.ibm.com/topics/knn))​​ ([Elastic](https://www.elastic.co/what-is/knn" \t "_blank))​.

### **Manhattan Uzaklığı (Manhattan Distance):**

ekran görüntüsü, top içeren bir resim

Açıklama otomatik olarak oluşturuldu

Manhattan uzaklığı, iki nokta arasındaki dikdörtgensel ızgara üzerinden hareket ederken alınan toplam mesafeyi ölçer. Bu ölçüm, taksi geometrisi olarak da bilinir. İki nokta arasındaki Manhattan uzaklığı şu şekilde hesaplanır: 𝑑(𝑥,𝑦)=∑𝑖=1𝑛∣𝑥𝑖−𝑦𝑖∣*d*(*x*,*y*)=∑*i*=1*n*​∣*xi*​−*yi*​∣ Bu uzaklık ölçüsü, genellikle şehir blokları arasında yolculuk yapmayı simüle etmek için kullanılır​ ([IBM - United States](https://www.ibm.com/topics/knn))​​ ([FreeCodeCamp](https://www.freecodecamp.org/news/k-nearest-neighbors-algorithm-classifiers-and-model-example/" \t "_blank))​.

### **Minkowski Uzaklığı (Minkowski Distance):**

çizgi, metin, daire, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Minkowski uzaklığı, Öklidyen ve Manhattan uzaklıklarını genelleyen bir uzaklık ölçüsüdür. p parametresi, uzaklığın hangi türde hesaplanacağını belirler: 𝑑(𝑥,𝑦)=(∑𝑖=1𝑛∣𝑥𝑖−𝑦𝑖∣𝑝)1/𝑝*d*(*x*,*y*)=(∑*i*=1*n*​∣*xi*​−*yi*​∣*p*)1/*p* Burada p = 1 olduğunda Manhattan uzaklığı, p = 2 olduğunda ise Öklidyen uzaklığı elde edilir. Farklı p değerleri, farklı türde metrikler oluşturur​ ([IBM - United States](https://www.ibm.com/topics/knn))​​ ([Elastic](https://www.elastic.co/what-is/knn" \t "_blank))​.

### **Chebyshev Uzaklığı (Chebyshev Distance):**

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Chebyshev uzaklığı, iki nokta arasındaki en büyük koordinat farkını ölçer. Bu uzaklık, satranç tahtasındaki bir kralın hamlelerini simüle etmek için kullanılır. İki nokta arasındaki Chebyshev uzaklığı şu şekilde hesaplanır: 𝑑(𝑥,𝑦)=max⁡𝑖(∣𝑥𝑖−𝑦𝑖∣)*d*(*x*,*y*)=max*i*​(∣*xi*​−*yi*​∣) Bu uzaklık, tüm koordinat farkları arasındaki maksimum değeri temsil eder ve özellikle maksimum toleransın önemli olduğu durumlarda kullanılır​ ([Elastic](https://www.elastic.co/what-is/knn" \t "_blank))​.

Bu uzaklık ölçümleri, KNN algoritmasının esnekliğini ve geniş uygulama yelpazesini artırır. Her ölçüm, belirli veri türlerine ve problem yapılarına uygun farklı avantajlar sunar.

## 

# BULGULAR

Proje sonucunda elde edilen bulgular, kullanılan modellerin meme kanseri teşhisinde etkili olduğunu göstermektedir. KNN modeli, yüksek doğruluk ve F1 skoru ile başarılı bir performans sergilemiştir. K-Means algoritması, verileri iki belirgin kümeye ayırmış ve doğru sınıflandırmada yardımcı olmuştur. SGDClassifier modeli ise büyük veri setleri üzerinde hızlı ve etkili bir şekilde çalışarak doğrusal sınıflandırma yapmıştır.

Aşağıda ise meme kanserinde K-NN modeli kullanarak elde ettiğimiz bulgular yer almaktadır.

### **Karışıklık Matrisi (Confusion Matrix):**

metin, ekran görüntüsü, dikdörtgen, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Karışıklık matrisi, modelin test veri setindeki performansını detaylıca gösterir. Grafikteki her bir hücre, modelin tahmin sonuçlarını ve gerçek sınıflarını ifade eder. Hücrelerin anlamları şu şekildedir:

**True Positives (TP)**: Modelin doğru bir şekilde 'malignant' (kötü huylu) olarak sınıflandırdığı örnekler.

**True Negatives (TN)**: Modelin doğru bir şekilde 'benign' (iyi huylu) olarak sınıflandırdığı örnekler.

**False Positives (FP)**: Modelin 'malignant' (kötü huylu) olarak yanlış sınıflandırdığı 'benign' (iyi huylu) örnekler.

**False Negatives (FN)**: Modelin 'benign' (iyi huylu) olarak yanlış sınıflandırdığı 'malignant' (kötü huylu) örnekler.

Bu matrisi kullanarak modelin doğruluğunu, hassasiyetini, geri çağırma oranını ve F1 skorunu hesapladık. Aşağıda bunların sonuçlarını görebilirsiniz:

### **Hesaplanan Değerler ve Anlamları:**

Karışıklık matrisi kullanılarak çeşitli performans metrikleri hesaplanabilir. Kodumuzun çıktısında bu metrikler şu şekildedir:

1. ***Accuracy (Doğruluk):*** Modelin tüm doğru tahminlerinin oranıdır.

Accuracy=𝑇𝑃+𝑇𝑁𝑇𝑃+𝑇𝑁+𝐹𝑃+𝐹𝑁Accuracy=*TP*+*TN*+*FP*+*FNTP*+*TN*​

Örneğimizdeki doğruluk oranı: 0.94 (Yani, %94 doğruluk oranı).

1. ***Precision (Hassasiyet):*** Modelin pozitif olarak tahmin ettiği örneklerin ne kadarının gerçekten pozitif olduğunun oranıdır.

Precision=𝑇𝑃𝑇𝑃+𝐹𝑃Precision=*TP*+*FPTP*​

Örneğimizdeki hassasiyet oranı: 0.98 (Yani, malignant olarak tahmin edilen örneklerin %98'i gerçekten malignant).

1. ***Recall (Geri Çağırma):*** Gerçek pozitif örneklerin ne kadarının doğru bir şekilde pozitif olarak tahmin edildiğinin oranıdır.

Recall=𝑇𝑃𝑇𝑃+𝐹𝑁Recall=*TP*+*FNTP*​

Örneğimizdeki geri çağırma oranı: 0.87

(Yani, gerçekten malignant olan örneklerin %87'si doğru bir şekilde malignant olarak tahmin edilmiş).

1. ***F1 Score:*** Precision ve Recall'un harmonik ortalamasıdır. Hem precision hem de recall'u dikkate alarak modelin genel performansını özetler.

F1 Score=2×Precision×RecallPrecision+RecallF1 Score=2×Precision+RecallPrecision×Recall​

Örneğimizdeki F1 skoru: 0.92

**Özet**:

Karışıklık matrisi ve bu matrise dayalı metrikler, modelimizin doğruluğunu ve sınıflandırma yeteneğini değerlendirmemize yardımcı olmaktadır. Özellikle dengesiz veri kümelerinde (örneğin, bir sınıfın diğerine göre çok daha fazla örneği varsa) bu metrikler, modelin performansını anlamada oldukça önemlidir. Projemizde elde edilen sonuçlara dayanarak, modelimizin yüksek doğruluk, hassasiyet ve geri çağırma oranlarıyla başarılı bir performans gösterdiği söylenebilir.

### **Doğruluk Grafiği (Accuracy Graph):**

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

KNN (K-En Yakın Komşu) algoritmasında, **k** parametresi, sınıflandırma kararını vermek için dikkate alınan en yakın komşuların sayısını belirtir. Farklı **k** değerleri, modelin nasıl performans göstereceğini etkiler. Doğruluk grafiği, bu farklı **k** değerlerine göre modelin doğruluğunu hem eğitim setinde hem de test setinde gösterir.

Grafikte, **k** değeri yatay eksende, doğruluk oranları (accuracy) ise dikey eksende gösterilmiştir. Eğitim seti doğruluğu ve test seti doğruluğu iki ayrı çizgi ile ifade edilmiştir.

* **Eğitim Seti Doğruluğu (Train Accuracy)**:
  + **k** değeri küçükken (örneğin, **k=1**), model eğitim setinde çok yüksek doğruluk gösterir. Bu, modelin eğitim verisine aşırı uyum (overfitting) gösterdiğini ve her bir eğitim örneğini neredeyse mükemmel bir şekilde öğrendiğini gösterir.
  + **k** değeri arttıkça, eğitim doğruluğu genellikle azalır çünkü model daha genel bir yapıya kavuşur ve eğitim setindeki küçük ayrıntılara aşırı uyum göstermemeye başlar.
* **Test Seti Doğruluğu (Test Accuracy)**:
  + **k** değeri çok küçük olduğunda, test doğruluğu genellikle daha düşük olabilir. Bu, modelin test verisinde aşırı uyum (overfitting) nedeniyle iyi genelleme yapamadığını gösterir.
  + **k** değeri arttıkça, test doğruluğu genellikle artar ve belli bir noktada en yüksek değerine ulaşır. Bu **k** değeri, modelin eğitim verisine yeterince uyum sağladığı ve aynı zamanda test verisinde iyi genelleme yaptığı anlamına gelir.
  + Ancak, **k** değeri çok fazla arttığında (örneğin, **k=20**), test doğruluğu tekrar düşebilir. Bu durumda, model yetersiz uyum (underfitting) göstermeye başlar çünkü çok fazla komşuyu dikkate alır ve karar sınırları belirsizleşir.

#### **Optimum k Değerinin Belirlenmesi:**

Grafikte, hem eğitim hem de test doğruluğunun iyi olduğu **k** değerini bulmak önemlidir.

En yüksek test doğruluğuna sahip **k** değeri, genellikle optimum **k** değeridir.

Grafikte test doğruluğunun belirli bir **k** değerinden sonra düşmeye başladığını gözlemleyebiliriz. Bu noktada, daha küçük **k** değerlerine odaklanmak mantıklıdır.

### **Özet:**

Doğruluk grafiği, KNN modelinin performansını ve uygun **k** değerini seçmek için kritik bir araçtır. Eğitim doğruluğu ve test doğruluğu arasındaki dengeyi sağlamak, modelin genelleme yeteneğini optimize etmek için önemlidir. Bu grafikte, eğitim seti doğruluğu ve test seti doğruluğu arasındaki farkın minimum olduğu ve test doğruluğunun en yüksek olduğu **k** değeri, genellikle modelin en iyi performans gösterdiği değerdir. Bu durum, modelin hem eğitim verisine yeterince uyum sağladığını hem de yeni veriler üzerinde iyi genelleme yaptığını gösterir.

Bu grafikte elde edilen sonuçlara dayanarak, **k=2** değeri modelimiz için oldukça iyi bir seçim gibi görünmektedir. Hem eğitim hem de test doğruluğu yüksek seviyelerde ve aralarındaki fark nispeten küçük, bu da modelin iyi bir genelleme yeteneğine sahip olduğunu gösterir.

### **Özellik Dağılım Grafikleri (Feature Distribution Graphs):**

### metin, diyagram, çizgi, plan içeren bir resim Açıklama otomatik olarak oluşturuldu

Özellik dağılım grafikleri, veri kümesindeki her bir özelliğin (feature) dağılımını görselleştirir. Bu grafikleri kullanarak, veri kümesindeki her bir özelliğin değerlerinin nasıl dağıldığını ve özellikler arasında herhangi bir dengesizlik olup olmadığını görebiliriz. Bu tür grafikler, veri analizi ve ön işleme aşamalarında oldukça faydalıdır.

#### **Grafiklerin Yorumu:**

***Histogramlar:***

Her bir histogram, belirli bir özelliğin değerlerinin frekans dağılımını gösterir. Yatay eksende özellik değerleri, dikey eksende ise bu değerlerin frekansları (kaç kez görüldüğü) gösterilir.

***Dağılımın İncelenmesi:***

Histogramlardaki çan eğrisi veya herhangi bir simetrik dağılım, o özelliğin yaklaşık olarak normal dağıldığını gösterebilir.

Özelliklerin bazıları belirli aralıklarda yoğunlaşmış olabilir, bu da veri setindeki belirli sınıfların bu özelliklerde belirli değer aralıklarında yoğunlaştığını gösterebilir.

Anormal derecede yaygın olan veya eksik olan değerler, veri setinde potansiyel outlier (aykırı değerler) veya eksik veriler hakkında bilgi verebilir.

**ÖZET:**

Özellik dağılım grafikleri, veri kümesindeki her bir özelliğin değerlerinin nasıl dağıldığını görselleştirir. Bu grafikler, veri analizi ve ön işleme aşamalarında kritik öneme sahiptir. Histogramlar, her bir özelliğin frekans dağılımlarını gösterir ve bu dağılımları inceleyerek veri kümesinde normal dağılım, yoğunlaşma alanları, outlierlar veya eksik veriler gibi önemli bilgileri tespit edebiliriz. Özellik dağılım grafikleri, modelin hangi özelliklere odaklanması gerektiğini anlamak ve veri setindeki dengesizlikleri belirlemek için kullanılır.

### **4. Sınıf Dağılım Grafiği (Class Distribution Graph): metin, ekran görüntüsü, diyagram, dikdörtgen içeren bir resim Açıklama otomatik olarak oluşturuldu**

Sınıf dağılım grafiği, veri kümesindeki hedef değişkenin (class) dağılımını gösterir. Bu grafik, veri kümesindeki her bir sınıfın (benign ve malignant) kaç örneğe sahip olduğunu görselleştirir.

#### **Grafiklerin Yorumu:**

***Countplot:***

Yatay eksende sınıflar (benign ve malignant), dikey eksende ise bu sınıfların sayısı gösterilir.

***Sınıf Dağılımı:***

Grafikteki çubukların yükseklikleri, veri kümesindeki her bir sınıfın örnek sayısını gösterir.

Dengeli bir dağılım, sınıflandırma modeli için daha ideal olabilir çünkü model her iki sınıfı da öğrenme ve doğru tahmin etme şansına sahiptir.

Eğer sınıflar arasında belirgin bir dengesizlik varsa (örneğin, benign örnekler malignant örneklere göre çok daha fazla ise), bu durum modelin performansını olumsuz etkileyebilir. Model, yaygın olan sınıfı daha iyi öğrenirken nadir olan sınıfı öğrenmede zorlanabilir.

### **Özet:**

Sınıf dağılım grafiği, veri kümesindeki hedef değişkenin (class) dağılımını görselleştirir. Bu grafik, sınıflar arasındaki dengesizliği tespit etmek ve modelin performansını değerlendirmek için önemlidir. Dengeli bir dağılım, modelin her iki sınıfı da doğru öğrenmesine yardımcı olurken, dengesiz bir dağılım modelin performansını olumsuz etkileyebilir. Çubuk grafikler, her bir sınıfın örnek sayısını gösterir ve veri kümesindeki genel yapıyı anlamamıza yardımcı olur. Sınıf dağılım grafiği, modelin dengeli bir şekilde eğitilip eğitilmediğini değerlendirmek için kullanılır.

# KAYNAKLAR

1. Arslan, E. (2021). "Makine Öğrenmesi: KNN (K-Nearest Neighbors) Algoritması." Medium. [Link](https://arslanev.medium.com/makine-%C3%B6%C4%9Frenmesi-knn-k-nearest-neighbors-algoritmas%C4%B1-bdfb688d7c5f)
2. Cover, T. M., & Hart, P. E. (1967). "Nearest neighbor pattern classification." IEEE Transactions on Information Theory, 13(1), 21-27. [Link](https://ieeexplore.ieee.org/document/1053964)
3. Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2004). "KNN model-based approach in classification." OTM Confederated International Conferences "On the Move to Meaningful Internet Systems." Springer, Berlin, Heidelberg. [Link](https://link.springer.com/chapter/10.1007/978-3-540-30465-2_12)
4. Kang, S. (2021). "k-Nearest Neighbor Learning with Graph Neural Networks." Mathematics, 9(8), 830. [MDPI](https://www.mdpi.com/2227-7390/9/8/830)
5. Kumar, P., & Suresh, A. (2017). "K-Nearest Neighbor Algorithm and its Applications." International Journal of Advanced Research in Computer Science, 8(3), 601-605. Link
6. Larose, D. T., & Larose, C. D. (2014). "k-Nearest Neighbor Algorithm." In Discovering Knowledge in Data: An Introduction to Data Mining (pp. 90-92). John Wiley & Sons.
7. Medium Türkiye (2019). "KNN: K En Yakın Komşu." Medium. [Link](https://medium.com/machine-learning-t%C3%BCrkiye/knn-k-en-yak%C4%B1n-kom%C5%9Fu-7a037f056116)
8. Smith, A. (2024). "K-Nearest Neighbors Algorithm: An Overview and Applications." OpenAI Resources.
9. Veritabanı Eğitimleri (2021). "K-NN ve Örnek Uygulama." Veritabanı Eğitimleri. [Link](https://www.veritabaniegitimleri.com/2021/01/18/k-nn-ve-ornek-uygulama/)
10. Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., ... & Steinberg, D. (2008). "Top 10 algorithms in data mining." Knowledge and Information Systems, 14(1), 1-37. [Link](https://link.springer.com/article/10.1007/s10115-007-0114-2)
11. Zhang, Z., & Zhou, Z. (2012). "Ensemble learning via sampling techniques." Artificial Intelligence, 196, 1-27. [Link](https://www.sciencedirect.com/science/article/abs/pii/S0004370212001085)

**KAYNAK KOD ADRESİ**

1. **Furkan Arda Karabulut GITHUB:** **https://github.com/NursemaKaya**
2. **Bünyamin Kavraz GITHUB:** **https://github.com/BunyaminKavraz**
3. **Yusuf Yeşiltepe GITHUB:** **https://github.com/YusufYesiltepe**
4. **Sena Firdes Zeybek GITHUB:** **https://github.com/flyinhor**

**5-Nursema Kaya GITHUB: https://github.com/NursemaKaya**